METHOD AND SYSTEM FOR
RELIABLE MULTICAST DATA TRANSMISSION

Inventor: Shekhar Amlekar
                 160, Bhagwannagar,
                 Nagpur 440027, India
                 Citizenship: India

Assignee: Novell, Inc.
                 1800 South Novell Place
                 M/S PRV-F-331
                 Provo, Utah 84606

METHOD AND SYSTEM FOR
RELIABLE MULTICAST DATA TRANSMISSION

BACKGROUND

[0001]   This disclosure relates generally to multicast data transmission and, more specifically, to a method and a system for ensuring complete transmission of multicast files with minimum retransmissions.

[0002]   A computer network generally has at least one computer (a server) that provides services to other computers (clients) via the network. Traditionally, network communications have employed a unicast transmission model in which a separate connection is made between a server and each of its clients. This model works particularly well when each client has unique demands of the server. However, when each client is receiving the same data, multiple copies of the same information packet can flood the network, causing congestion and escalating bandwidth requirements.

[0003]   Multicasting may alleviate some of the problems associated with unicasting by enabling a server to transmit a single packet of information that may be received by multiple clients who have requested the information. Multicast enabled switches and routers permit the single packet of information to be copied so that only a single packet

need be forwarded through each branch of the network to reach the clients requesting the information. Although multicasting may be used in a wide variety of applications in which large amounts of information are being sent to multiple clients, multicasting generally used for applications in which many clients want the identical data simultaneously. Such applications may include live audio or video transmissions, multi-user games, and real-time stock tickers.

[0004]   Despite the benefits of multicasting for some applications, present implementations of multicasting do not enable a server to receive performance information back from clients, and therefore, transmission reliability may be sacrificed. Current methods of multicasting may also result in unrecoverable data loss.

[0005]   Therefore, what is needed is an improved method and a system for the complete transmission of multicast files. It is also desirable to maintain efficiency by minimizing retransmissions.

SUMMARY

[0006]   Provided is a method and system for reliably multicasting a data transmission from a server to one or more clients. The server and clients may be connected via a control channel and a multicast data channel. In one embodiment, the method comprises sending a first data transmission to the clients over the multicast data channel and receiving a response over the control channel from at least one of the clients identifying data not received by the client during the first data transmission. A minimum retransmission data set is determined based on the response, wherein the minimum retransmission data set includes at least a portion of the data not received by the client during the first data transmission. The minimum retransmission data set is sent over the multicast data channel.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 is a flowchart illustrating one embodiment of a method for data multicasting in the network of Fig. 1.

[0008] Fig. 2 is a diagram of an exemplary computer and network environment in which the methods of Figs. 2, 3a, and 3b may be executed.

[0009] Fig. 3a is a flowchart illustrating another embodiment of a method for data multicasting in the network of Fig. 1.

[0010] Fig. 3b is a flowchart illustrating a file reception process of Fig. 3a.

[0011] Fig. 4 is a scorecard that may be used with the method of Figs. 3a and 3b.

DETAILED DESCRIPTION

[0012] This disclosure relates generally to multicast data transmission and, more specifically, to a method and a system for ensuring complete transmission of multicast files with minimum retransmissions. It is understood, however, that the following disclosure provides many different embodiments or examples. Specific examples of components and arrangements are described below to simplify the present disclosure. These are, of course, merely examples and are not intended to be limiting. In addition, the present disclosure may repeat reference numerals and/or letters in the various examples. This repetition is for the purpose of simplicity and clarity and does not in itself dictate a relationship between the various embodiments and/or configurations discussed.

[0013] Referring to Fig. 1, in one embodiment, a method 100 enables a computer (e.g., a server) to efficiently provide identical information to a plurality of other computers (e.g., clients). As will be described in greater detail in Fig. 2, the server may be connected to the clients via a control channel and a data channel. In step 102, the server transmits one or more files over the data channel using a multicast transmission. In step 104, the server receives a response from the clients over the control channel. If a

file is not received in its entirety by one or more clients, the response may identify

which parts of the file were not received by each client. If a client received the entire

file, it may either not respond to the server or may indicate that the entire file was

received, depending on the particular implementation of the method 100. For example,

if a client chooses not to acknowledge the successful reception of the file, the lack of

acknowledgement may be interpreted as an acknowledgement by the server after a

predefined amount of time has expired with no further communication from the client.

[0014]   In step 106, a determination is made (based on step 104) as to whether each

file was received by each client in its entirety. If each file was received, the method 100

ends. However, if each file was not received, the method 100 continues to step 108,

where the server determines a minimum retransmission data set based on the response

of step 104. In the present example, the minimum retransmission data set includes

each part of the file that was not received by the clients. In step 110, the server then

sends the minimum retransmission data set to the clients. The minimum

retransmission data set may be sent to all the clients or only to the clients who

indicated that they were missing data. The method 100 then returns to step 106. Steps

106-110 may be repeated until all clients have received the data transmitted in step 102.

[0015]   Referring now to Fig. 2, an exemplary communications system 200, within

which the method 100 of Fig. 1 may be executed, is illustrated. The communication

system 200 includes a computer network 202. Connected to the computer network 202

are a server 204 and one or more client computers 206 ("clients"). Additional servers,

which may include host and proxy servers, may be connected to the computer network

202 to expand the services provided by the communications system 200. The computer

network 202 may be, for example, the public Internet or a private intranet, and may

include a plurality of routers 208, switches 210, and/or other equipment for connecting

the server 204 to the clients 206. Further, the computer network 202 may include, for

example, a satellite network, a terrestrial network, or a combination of the two. The

routers 208 and switches 210 may be multicast enabled. It is understood that, in some

embodiments, the client computers 206 may act as a server to other client computers 206.

[0016] The server 204 and the clients 206 may be connected to the network by a control channel 212 and a data channel 214. As one example, the communications system 200 may utilize a satellite link as the data channel 214 and may use a terrestrial connection as the control channel 212. The control channel, which may use a protocol such as a transmission control protocol (TCP), may be used to send meta-data that provides descriptive information about a file or application being transmitted. Examples of such descriptive information may include attribute information such as name, size, and data type, as well as structural information such as length of fields, location of data, and how the data is associated with other data. The control channel 212 may remain open across multiple file transfer sessions to avoid the need for repeated connection establishment and termination.

[0017] The data channel 214 may utilize one or more of a variety of multicast protocols for multicast data delivery. To conserve network resources, the data channel 214 may remain open only during the transmission of data and may be closed after a transmission has ended. The multicast protocol selected may be, for example, a pragmatic general multicast (PGM) protocol as specified in the Internet Engineering Task Force (IETF) Request for Comment (RFC) 3208 entitled, "PGM Reliable Transport Protocol Specification." The PGM protocol may enable a receiver of a multicast session to detect unrecoverable data loss. Unlike unicast protocols such as TCP, which utilize positive acknowledgments (ACKs) to guarantee reliable data delivery, PGM uses a negative acknowledgment (NAK) and negative acknowledgment confirmation (NCF) system to alert the server when data is not delivered. Unicast protocol ACKs may add excessive traffic to a network, especially for sessions transmitted from one server to many clients. By comparison, PGM does not contribute heavily to network congestion because NAKs are only generated when data is not received, and NCFs limit redundant NAKs. However, PGM does not prevent a client from experiencing

unrecoverable data loss, nor does PGM provide a mechanism for minimizing the retransmission of lost data.

[0018]   Referring now to Fig. 3a, in another embodiment, a method 300 enables reliable data multicasting from a server to one or more clients with minimum retransmissions. The method begins, at step 302, when a client (e.g., the client 206 of Fig. 2) communicates an interest in receiving a transmission from a server (e.g., the server 204 of Fig. 2), thereby establishing a control channel (e.g., the control channel 212 of Fig. 2) between the client and the server. A transmission may include multiple files, and the files may include multiple data packets. At step 304, the client 206 is enrolled into a scoreboard (Fig. 4) used by the server 204 for file transmission.

[0019]   With additional reference to Fig. 4, a scoreboard 400 includes an identity 402 of each client 206. The identity may be, for example, an internet protocol (IP) address or other identifier associated with the client 206, such as a media access control (MAC) number. Upon enrollment, the client 206 may become a member of a permanent transmission group 404 which includes clients who have expressed an interest in receiving the file transmissions by establishing a control channel connection. In the present example, the Boolean character "1" indicates membership in the group, and the character "0" indicates that a particular client's IP address is not a member of the group. Initially, the client 206 may also become a member of a current transmission group 406. Membership in the current transmission group 406 may be indicated using the Boolean scheme described above. The current transmission group 406 may include clients 206 toward whom the server's current session is directed. The server 204 may periodically scan the control channel 212 for each client 206 to confirm client connectivity.

[0020]   Referring again to Fig. 3a, at step 306, all members of the permanent transmission group 404 (Fig. 4) are made members of the current transmission group 406 (Fig. 4). The server 204 may confirm that all members of the current transmission group 406 are connected using the scoreboard 400 or another method. At step 308, the server 204 may announce the upcoming transmission over the control channel 212 to

all permanent transmission group members. The announcement may include the number of files to be transferred, a filename or filenames, the number of byte ranges to be transmitted for each filename, and the starting and ending bytes for each of the byte ranges to be transmitted in the session. Proceeding to step 310, the clients 206 comprising the current transmission group 406 may begin listening on a data channel (e.g., the data channel 214 of Fig. 2). At step 312, the server 204 begins the transmission of the file over the data channel 214, and at step 313 the process of reliable file reception begins.

[0021]   Referring now to Fig. 3b, a method 314 for reliable file reception begins at step 316 with the clients 206, which are members of the current transmission group 406, receiving the transmission over the data channel 214. Proceeding to step 318, if the file is received in its entirety by each client 206, the client proceeds to step 322 and continues to receive the additional files in the transmission. Because the client 206 has received the announcement as described in step 308, the client may be aware of the expected bytes to be delivered. The client 206 may compare the expected byte ranges to the received byte ranges to determine whether the file was received in its entirety. If the file is incomplete, the client 206 records the bytes corresponding to the missing data packets at step 320 and continues receiving the transmission at step 322. If, for example, PGM is the protocol utilized for the data channel, missing data packets that cannot be restored cause the client 206 to detect an unrecoverable data loss. PGM allows the client 206 to record information about the missing data packets using an option of the protocol, OPT_FRAGMENT. This option allows the client 206 to know the amount of missing data and the byte off-set where that data packet belongs in the file, while continuing to receive transmissions.

[0022]   At step 324, the client 206 may detect the end of the transmission by receiving an end-of-file (EOF) indicator from the server 204. The client 206 then closes the session on the data channel 214, and at step 326, communicates to the server 204 over the control channel 212 the missing byte ranges not received in the data transmission of step 312. The communication may include the client's IP address, the number of files

to be transferred, the file name or names, the number of missing byte ranges for each filename, and the starting and ending byes for the missing byte ranges. The communication from the client 206 to the server 204 over the control channel 212 may also include any ACKs or NAKs required by the multicast protocol. If, at step 328, the client 206 has received the transmission in its entirety, the missing byte range may be indicated as NULL, which may serve as an acknowledgement to the server 204 that the transmission was received. If the missing byte range is NULL, the client 206 may be removed from the current transmission group 406 at step 330, but at step 332 may continue listening on the control channel 212 for announcement of the next transmission session. If, at step 328, the missing byte range is something other than NULL, the retransmission process begins.

[0023] At step 334, the server 204 aggregates the missing byte ranges received from all clients 206, and determines the minimum superset of the missing byte ranges for retransmission. At step 336, the server 204 announces this minimum superset for retransmission over the control channel 212. Responsive to the announcement over the control channel 212, a client 206 listening on the control channel and interested in the retransmission may select whether to reestablish a data channel 214 connection at step 338. The current transmission group 406 of the scorecard 400 is updated to include the clients 206 which have reestablished the data channel 214 for the retransmission. Because only the interested clients 206 connect for the retransmission based upon the announcement over the control channel 212, data may not be carried to clients 206 that are not interested in the data. The server 204 may then, at step 340, begin transmitting at least a portion of the original transmission, which may be the minimum superset of the missing byte ranges, to the members of the current transmission group 406 over the established data channel 214. At step 342, the clients 206 receiving the retransmission select from the retransmission only the bytes which were not received in the prior transmission. The steps 318 to 342 may be repeated until no clients exist in the current transmission group 406.

[0024] The methods 300 and 314 may be used for the multicast transmission of a variety of data. The transmission can, for example, include music files, financial data, movie videos, support packs, security bug fixes in a network management system, web content, or other content of interest to multiple clients or desired by multiple clients simultaneously.

[0025] The methods described in Figs. 3a and 3b may also be used to accommodate a client who joins the permanent transmission group 404 after a transmission has begun. The client may be made a member of the current transmission group 406 and may receive an announcement for the session over the control channel. This announcement may include a filename, the number of byte ranges, and the starting and ending bytes for each of the byte ranges to be transmitted in the session. When the client connects to the server on a data channel and begins receiving the transmission of the session, the client may know and record the missing bytes based upon the control channel announcement. The client may then recover the missing bytes as described in Fig. 3b.

[0026] If a client disconnects during a transmission, the client may be removed from both the permanent transmission group and current transmission group. If a client disconnects while not a member of a current transmission group, the client may still be removed from the permanent transmission group.

[0027] The methods 300 and 314 may also be used for multicasting in selected tiers of a tiered electronic distribution (TED) application which allows one server to distribute data files, server applications, or other types of data to multiple subscribing servers. Still another application for the methods 300 and 314 is reliable content delivery within a content distribution network (CDN). CDNs provide a method of alleviating network congestion by caching and serving web site content from servers near the client rather than from the server of origin. The methods 300 and 314 may also be used to transmit multicast data from a central office to a branch office using a branch office management appliance.

[0028] The methods 300 and 314 may also incorporate forward error correction (FEC) software which may reduce the number of retransmissions by alleviating some

portion of the data packet loss. FEC software may be used with the original transmission and/or may be used with subsequent retransmissions.

[0029] While the preceding description shows and describes one or more embodiments, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the present disclosure. For example, the type of protocols used in the preceding description may vary, and it is understood that substitutions may be made. Similarly, different network configurations may be used for different types of digital devices. Therefore, the claims should be interpreted in a broad manner, consistent with the present disclosure.